Nom : ..... Prénom : .....

2B1 Devoir d'informatique III 30.06.2014

Question 1 20 (=6+6+8) points

Pour implémenter les polynômes, on définit le type `poly` par :

```
type poly = record
              nom:string;
              c:array[0..100] of extended;
              d:integer
            end;
```

où le champ `nom` contient le nom, le champ `d` le degré et le champ `c` la liste des coefficients du polynôme suivant les puissances croissantes de la variable.

(1) Ecrire la fonction `horner` qui prend en entrée un polynôme `p` de type poly et un réel `x` de type extended et qui retourne la valeur numérique du polynôme `p` en `x` à l'aide de l'algorithme de Horner.

(2) Faire un tableau d'exécution de la fonction `horner` montrant comment elle calcule $p(-3)$, lorsque $p(x) = 6x^4 + 7x^3 - 5x + 8$.

(3) a) Ecrire toutes les ***affectations*** qui permettent de stocker le polynôme

$$a(x) = 1 - 2x + 3x^2 - 4x^3 + ... - 20x^{19} + 21x^{20}$$

dans une variable `a` de type `poly`. Les coefficients du polynôme doivent être définis à l'aide d'une boucle. *N.B. : Il ne s'agit pas de demander à l'utilisateur d'entrer les coefficients à l'écran, donc votre réponse ne doit contenir ni* `writeln`, *ni* `readln` ! b) Ecrire ensuite l'instruction qui permet de calculer et d'afficher à l'écran $a(3 - a(2))$ en utilisant la fonction `horner`.

Question 2 10 points

Pour implémenter les fractions, on définit le type `fraction` par :

```
type fraction = record
                  n:integer;
                  d:integer;
                end;
```

où le champ `n` représente le numérateur et le champ `d` représente le dénominateur.

Ecrire la procédure `saisie` avec un paramètre `x` de type fraction qui demande à l'utilisateur d'entrer une fraction sous la forme a/b avec b<>0. Cette fraction sera stockée dans le paramètre `x`. ***Consignes*** à respecter : a) La procédure n'acceptera pas un dénominateur égal à 0. b) La procédure devra traiter correctement le cas où l'utilisateur entre un entier (donc par exemple 6 au lieu de 6/1).

Question 3 — 18 (=10+8) points

Pour implémenter les matrices à éléments entiers, on définit le type `matrice` par :

```
type matrice = record
                 nom:string;
                 items:array[1..100,1..100] of integer;
                 rowCount,colCount:integer;
               end;
```

Les champs `rowCount` et `colCount` contiennent respectivement le nombre de lignes et de colonnes de la matrice, le champ `items` contient les éléments et le champ `nom` contient le nom de la matrice.

(1) Ecrire la procédure `matriceAuHasard` qui permet de définir aléatoirement une matrice : a) le nom de la matrice est une lettre majuscule choisie au hasard entre `'A'` et `'Z'` b) le nombre de lignes et de colonnes doit être le même : c'est un entier aléatoire choisi entre 2 et 10 ; c) les élements de la matrice sont des entiers choisis aléatoirement parmi -1, 0 et 1 avec ***en moyenne*** la ***moitié*** des éléments égale à 0, un ***quart*** des éléments égaux à 1 et un ***quart*** égaux à -1.

(2) Ecrire la fonction `produit` qui calcule le produit de deux matrices données de type `matrice`. Rappeler la condition nécessaire pour que le produit existe. On n'aura pas besoin de tester dans la programmation si cette condition est vérifiée.

Question 4 — 12 (=3+9) points

Pour implémenter les triangles du plan euclidien, on définit les types `point` et `triangle` respectivement par :

```
type point = record
               x,y:extended;  // x = abscisse, y = ordonnée
             end;

  triangle = array[1..3] of point;
```

(1) Ecrire la fonction `sqrLongSegment` qui prend en entrée deux points `A` et `B` et qui retourne le ***carré de la longueur*** du segment [AB].

(3) Ecrire la fonction `triRectangle` avec un paramètre `t` de type `triangle` et qui retourne `true` si ce triangle est rectangle, `false` sinon. ***Consigne à respecter*** : les carrés des longueurs des 3 côtés devront être stockés dans un tableau de type `array` (variable locale de la fonction) et on recherchera le côté le plus long (hypoténuse).

G. Lorang